

A Project Report

(Review III report)

On

COURSE REGISTRATION SYSTEM

Submitted by

Riyesh Duwal Shrestha-20BCE2782

Karma Gurung-20BCE2933

Divit Hitul Popat-20BCT0319

Tarun RM -20BCE2262

Sai Nikshith G -20BCE2758

For

Software Engineering

CSE3001

Slot: B1, J Component

B.Tech. in Computer Science and Engineering



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

November 2022

Table of Contents

Chapter 1 Introduction to Project

- 1.1 Introduction
- 1.2 Problem Definition
- 1.3 Project Scope
- 1.4 Motivation
- 1.5 Background Study/Literature Survey
- 1.6 SDLC approach used to develop project

Chapter 2 Project Planning

- 2.1 Project Schedule
- 2.2 Effort and Resource Estimation

Chapter 3 Requirement Gathering and Analysis

- 3.1 SRS
- 3.2 Data Modelling
- 3.3 Structural Analysis

Chapter 4 Designs

- 4.1 UML Designs
- 4.2 Algorithms and Flow Charts

Chapter 5 Development

- 5.1 Tools Description and Development approach used
- 5.2 Pseudocodes of Important Modules.

Chapter 6 Testing

- 6.1 Test cases for Important Modules

Chapter 7 Screenshots of Developed Product

Annexures

- 1. Review I Presentation
- 2. Review II Presentation
- 3. Review III Presentation

List of Figures

List of Tables

Chapter 1 Introduction to Project

1.1 Abstract

The online course registration system is the central part of the educational administration system. With the advent of Information Technology in the last decade, the major focus has shifted from manual systems to computerised systems. Various systems viz. railway reservation, hospital management etc. involving manual work have been automated efficiently. Student course registration process in colleges involve filling registration forms manually, getting it signed by respective subject teachers, and then getting the documents acknowledged from the concerned Advisors, College Deans and Accounts Officers respectively. Finally the registration forms are submitted in the Administrative Branch. As is evident, this process is very laborious and time consuming. An Online Student Course Registration System has been developed to simplify the current manual procedure. This system has been developed using PHP, MongoDB.. This system software is more secured, user-friendly and less time-consuming. The new course registration system can be put into use by any educational institutes. By registering the course voluntarily, the new system improved registration mechanism, implemented course registration of common course for undergraduate and graduate students, and also supported the teaching activities across spring, summer and fall semesters. This article introduces the design and implementation of the new online course registration system, including registration mechanism, technical architecture, and system design, etc.

1.1 Introduction

The project objective will be focused on developing an online course registration to ensure the effectiveness of the flow of registration. Moreover the system will offer a complete management system that integrated with the online course registration to help the stakeholder for maintaining the flow process of the course. The registration process can be done online without the need of paperwork anymore. It is also help the student to get more information about the course process while they enrolled. System may be defined as a layered structure that depicts how programs involved would interrelate and communicate. In computers, System may also include actual programs, programming interfaces and tools for managing the larger system. The term system may be used differently in different contexts, but more or less the concept remains the same. Online student course registration system combines multiple systems to construct a combined framework. This framework consists of multiple modules, which further contain different systems along with the implementation of their defined constraints. Basically, systems are implemented for facilitating complex manual processes and that is exactly what we are trying to achieve. System is implemented as per user requirement such as a manufacturing concern may install a plant for easing out manual processes. We have sought help from computer programming for automation of manual registration system. With the introduction of computers, every aspect of our lives has been revolutionized. When used judiciously, computers can help us save time, secure our personal information, access the required information whenever and wherever required. Keeping all these positive points in mind, we have developed an Online Student Course Registration System for easily managing the semester registration process for the student in an institution. Ours is an advisory based system. In state agricultural universities the course allocation is advisory based and more complicated. The courses are assigned according to the skill set and industry requirements. Hence, in current scenario, automated system is required for course registration of students.

1.2 Problem Definition

Course Reservation System is an interface between the Student and the Registrar responsible for the issue of Course. It aims at improving the efficiency in the issue of Course and reduces the complexities involved in it to the maximum possible extent. If the entire process of 'Issue of Course' is done in a manual manner then it would takes several months for the course to reach the applicant. Considering the fact that the number of applicants for course is increasing every year, an Automated System becomes essential to meet the demand. So this system uses several programming and database techniques to elucidate the work involved in this process. It also has a notification system that notifies the user on their interested course viability.

1.3 Project Scope

- Course selection
- Course requirement validation
- Course modification
- Teacher selection
- Notification system

The scope of this project includes our group of developers assisted by our customer. The scope thus far has been the completion of the basic interfaces that will be used to build the system. The database to be used will be setup by the developers and given the necessary permissions. The online registration system will be used by students whom may be familiar or not to the online registration process thus the scope of the project must be user friendly for both our customers and administrators.

This system will allow only students to register, constraints will be put on the system, like prerequisites for all courses offered. Each student will indicate two alternative choices in case a course offering becomes filled or cancelled. Once the registration process is completed for a student, the registration system sends information to the billing system so the student can be billed for the semester.

1.4 Motivation

Software that will help the student to get the information about a particular course and then they can easily register themselves in a particular course. The management of the institution can easily see the records of the students, courses and fees. Also is designed to be interactive and helps the user to manage their online courses. Users will be able to register and login. A feedback system will be provided to know how the user feels about the courses. All the details of the course can be viewed before registering the course. The automatic notifications have the dual advantage of improving the user experience of the platform and simplify the management of training, as they reduce manual actions in every phase of the training process: from sending the credentials to access the course, to extracting and sending reports with the statistics of use, to the dissemination, collection and analysis of questionnaires of learning or satisfaction. Showing alerts and notifications of courses users are interested which data is taken in the registration form.

1.5 Background Study/Literature Survey

Online Registration

An online student registration system streamlines the application, registration, and monitoring of students in a school or training institute. According to Morris Wall (1990), online registration systems are websites that allow users to sign up for memberships, events and training by completing a form. Online registration systems are replacing manual processes, such as registering by telephone, mail or at events using paper forms. Many universities and educational institutions have a wide range of courses, students and faculty. Registering for classes online reduces paperwork, personnel, cost and conserves resources. It has been defined by Ralph E. Johnson (1996) as a system where many of you register for courses online. The computer knows all the courses that are available each semester and also knows which ones you are taking. It makes sure that a student can't register for too many courses, and that a course is not oversubscribed.

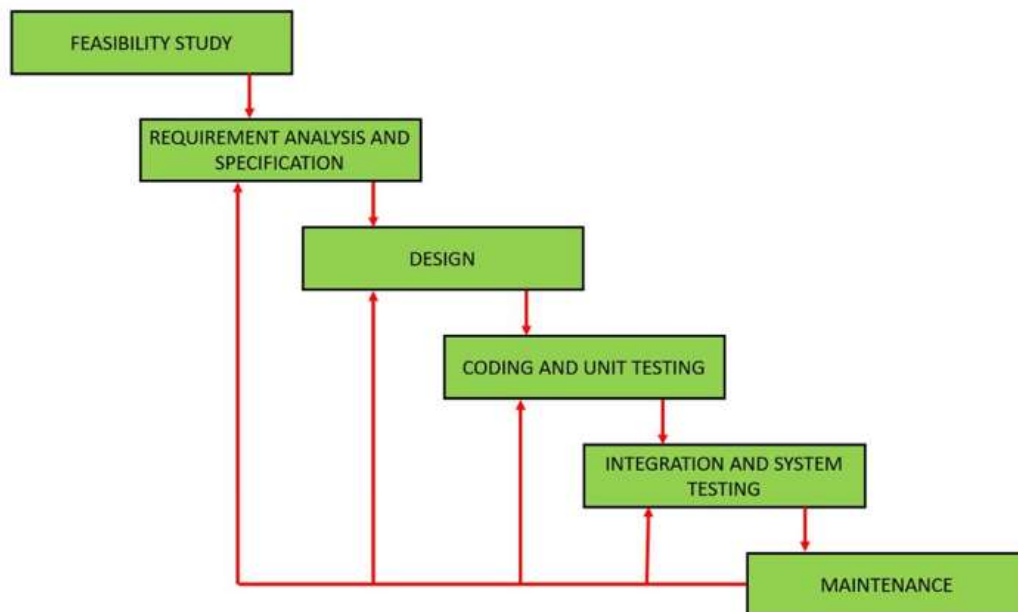
Existing Online Registration Systems

Online Registration University of Peradeniya

There are a number of online registration systems that have been made in different context and to suit the need arising. The online course registration for the faculty of engineering in university of Peradeniya and in their research they generated a solution named Online Registration. The project was divided into three namely, Online Registration which is the start-up project of their system, and depending on Data Access and Web security. Data Access would actuate the data transaction between the client and the server and Web Security would configure the database and handle the authentication and authorization. The research described their experience in designing, developing and deploying an online course registration system at the Faculty of Engineering in University of Peradeniya. The system has not only reduced the burden of all parties involved in the course registration process, but also improved the process by reducing errors. Though their system managed to reduce the errors significantly it lacked the continued interaction with the student in the updates of the student's registration status updates. It also lacked accountability as some stakeholders in the registration process do not have access to the system.

1.6 SDLC approach used to develop project

We will be using Iterative waterfall model to implement our project. We already know what our end goals are and we are very sure that the end goals won't change plus our project doesn't have dynamic needs. If incase of any change in requirement we can change accordingly in every stages. Risks will be very minimal in our project so we don't have to worry about model not being able to handle risks. Iterative Waterfall is also a simple model to implement.



Chapter 2 Project Planning

2.1 Project Schedule



2.2 Effort and Resource Estimation

Chapter 3 Requirement Gathering and Analysis

3.1 SRS

The online registration system comprises of two main features, namely, internet connectivity which will enables users to communicate with the server through a browser or web agent, and secondly the system requires database service to store the user's data. In a nutshell this system is web application and thus is only operational in an internet enabled environment.

St-1: Click "Register" Button: Account Registration

1. The system shall allow non-registered user to create a secure account.
2. The system shall require the following information: First name, Last name, E-mail address.
3. The system shall ask the user for student number as a username and password.
4. The system shall confirm the username and password are acceptable
5. The system shall store the information in the database

St-2: Click "Login" Button: Account Login

1. The system shall allow a registered user to log-in to their account.
2. The system shall require a username and password from the user.
3. The system will verify the username and password, and the user will be considered "logged-in".

St-3: Click "Registration" Button: Course Registration

1. The system shall allow users to register for courses offered.
2. The system shall require user's personal formation such as: Full name, Identity number, date of birth, Address, etc.
3. The system shall require the user to specify the level of study and allow the user to select courses to register for.

4. The system shall confirm all valid courses offered for the level of study selected by user.
5. The system shall also require the user to provide credit information.
6. The system shall allow user to save the already entered details for later continuation.
7. After the user filled the entire form, the system shall save the data in the database.

St-4: Click “Cancel Registration” Button: Cancel

1. The system shall allow the user to cancel the entire registration.
2. Alternately the user may wish to cancel while filling the form.

St-5: Click “Replace Course” Button: Replace

1. The system shall allow user to replace one with another.
2. This can happen while or after registration.
3. The system shall be given option to progress back to invalid courses entered.

St-6: Click “View available Courses” Button: View Courses

1. The system shall allow user to view available courses before registration.
2. The system shall view available space left for each course.

St-7: Click “submit registration” Button: submit

1. After filling in the registration form, the system shall allow user to submit the registration.
2. After submitting a copy of the proof of registration will appear on the screen.

St-8: Click “view registration status” Button: view status

1. System shall allow user to click and view their registration status. This includes the courses registered and proof of registration.
2. System shall allow user to print the proof of registration

Functional Requirements

REQ-1: The system shall be internet oriented and require an online server.

1. The online registration process shall be done online with the user requesting HTML documents through a browser using TCP/IP and the server shall use PHP script to manipulate the entered details and SQL to manipulate the data.
2. The server shall request response from the user for data saving and data requesting.

REQ-2: The system shall save the user's details to a remote database service

1. The user shall be allowed to enter account details which will be saved in a remote database server.

REQ-3: The system shall allow users to register and to log in a user account

1. The system will identify each user by a user account.
2. Each user shall be able to access the system through the account and therefore be able to register for courses.

REQ-4: The system will allow users to register for courses following the University Rules.

1. Before registering for courses, the users must be qualified for registration and the registration fee must have been paid.
 2. The pre-requisite courses shall be considered before registration.
-
1. The system shall allow users to view courses, the number of applicants and the space left for admission.
 2. This information must be updated by the administrator.

REQ-6: The system shall give out notification if any interested course are available

1. Users who did not get the registration of interest will get notified on interested course
2. They will be notified as available course.

REQ-7: The system will block courses for students whom failed prerequisites

1. If student failed prerequisites, they will be blocked to register courses that require the prerequisites.

Other Nonfunctional Requirements

Performance Requirements

The system is required a fair amount of speed especially while browsing through the catalogue and presenting different possibilities for the schedule. The outcomes of the product are not directly influenced by its speed, because all the operations are linked to each other and one operation cannot be computed before the one causing it and also since the availability of the internet connection is not predictable.

The database shall be able to accommodate a minimum of 10000 records of students.

The software will support multiple users, with their respective accounts of course.

Safety Requirements

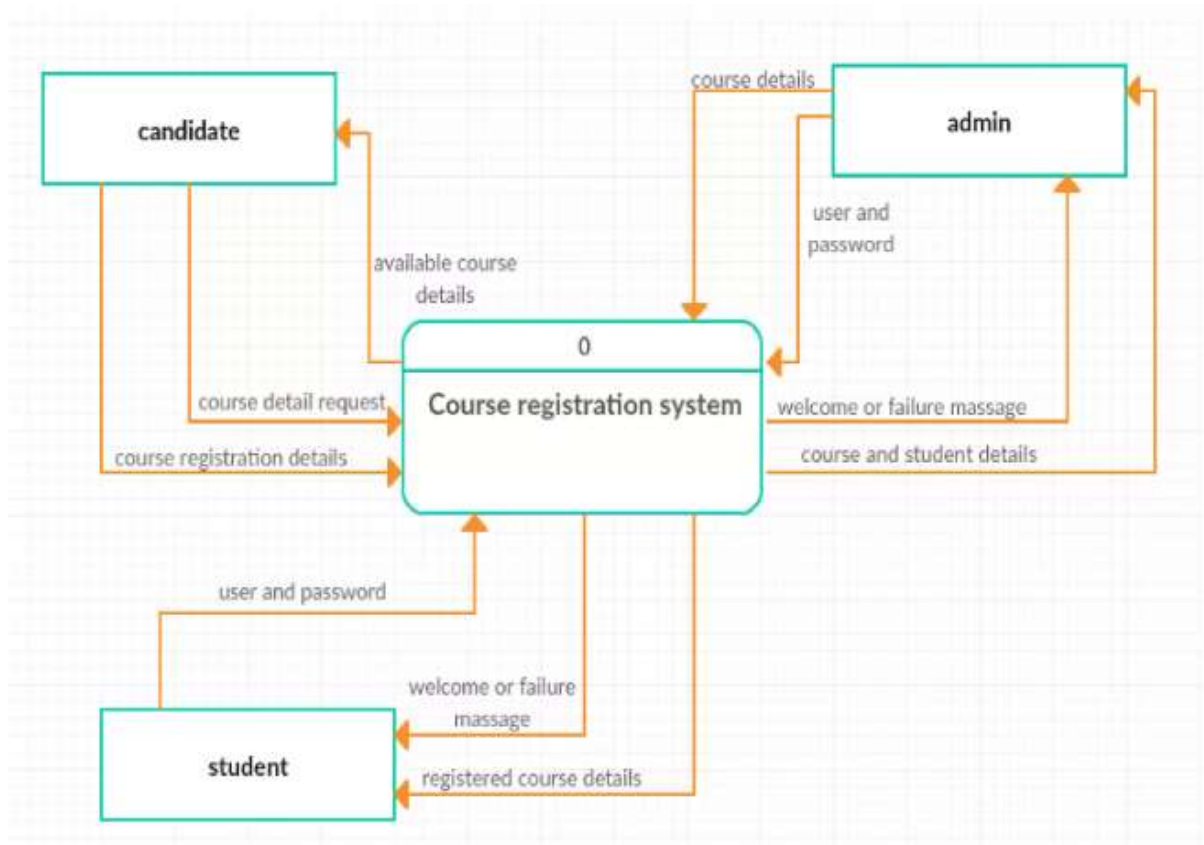
The system will provide a protection of the database such as the one that the university already provides. However, the system will have to increment this level of protection because of the personal data made available on the system, and the larger share of people that will be having access to it through the online registration.

The users' privacy will be granted by the limited access that the log-in process is going to give to the database. Also, the system does not grant direct access to the database itself. Stakeholders (group2) and developers (group4) who need to access the database will have to access it from a source independent from the registration system.

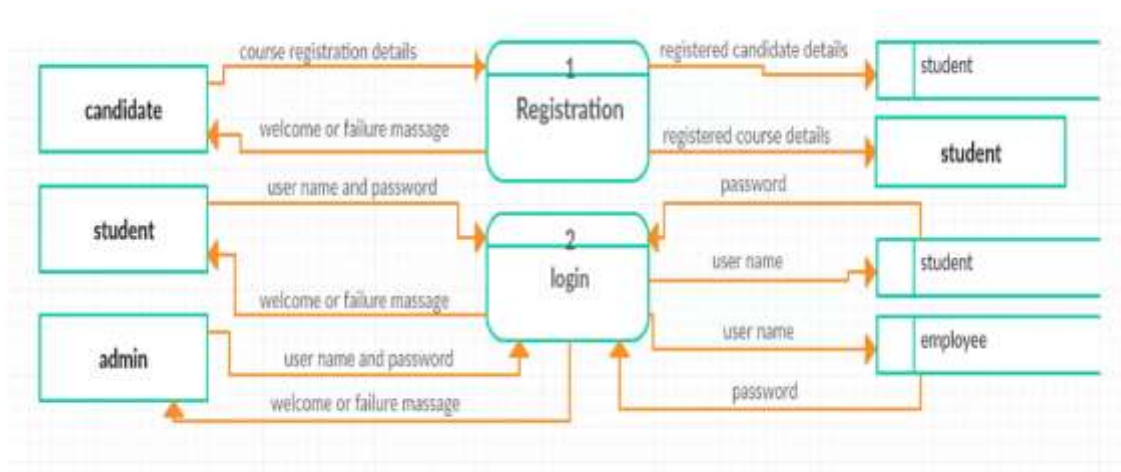
3.2 Data Modelling

Data flow diagram:

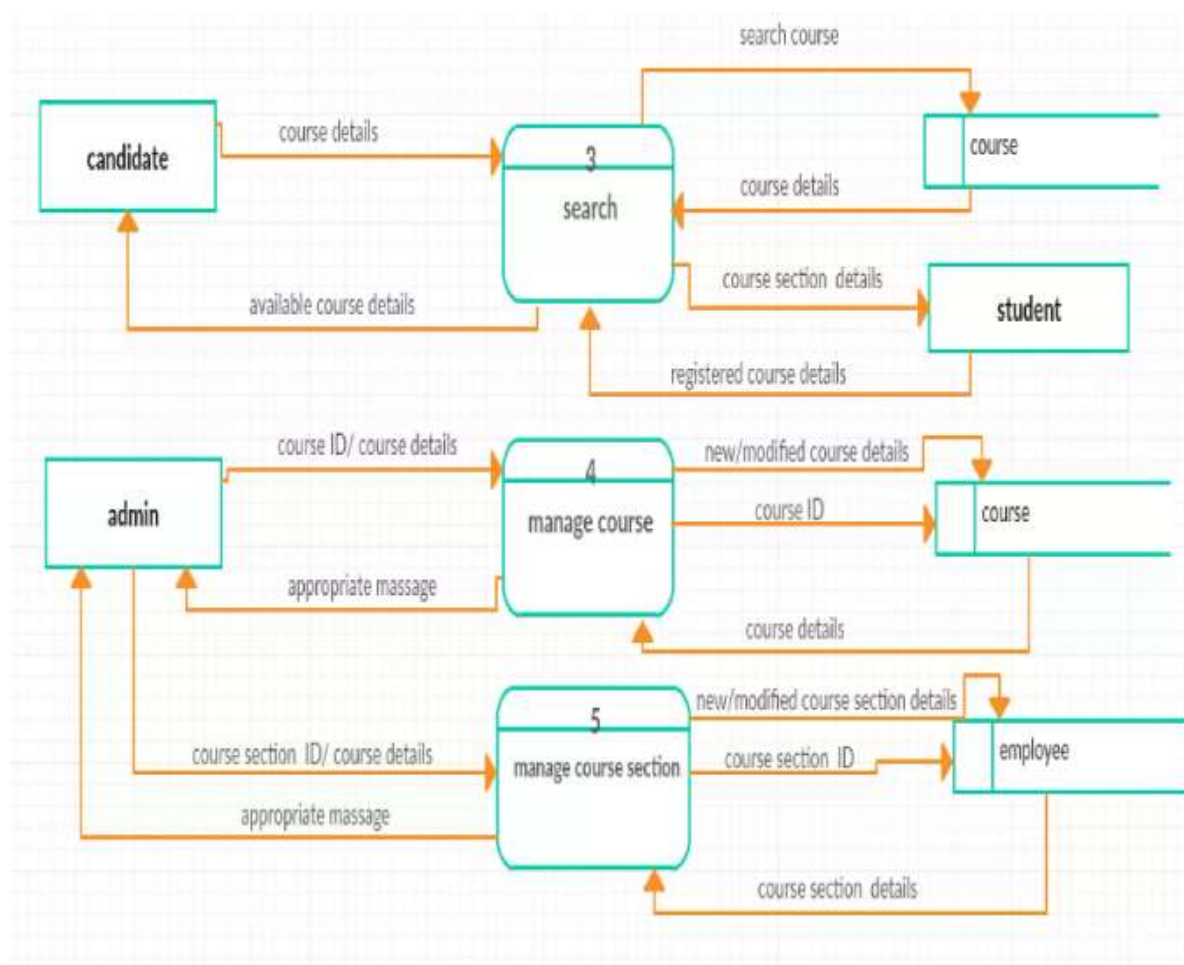
Context diagram:



DFD-1:

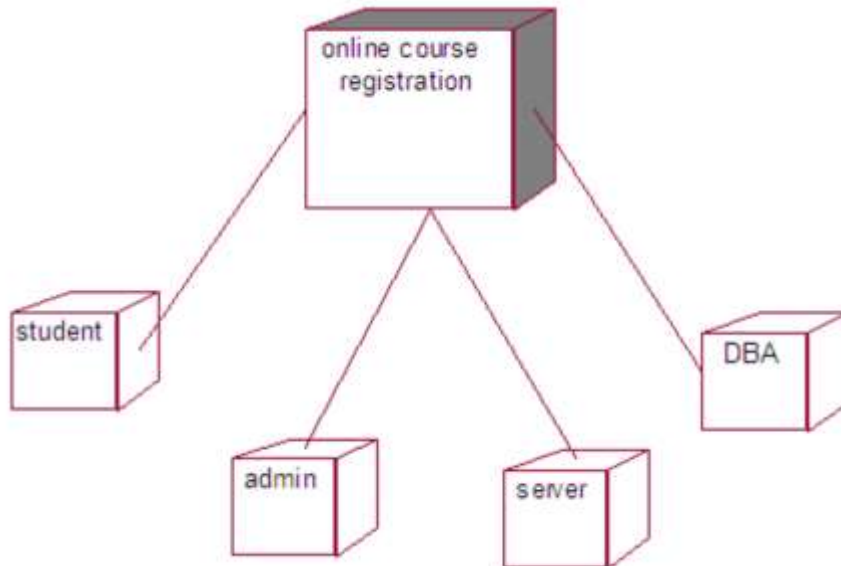


DFD-2:

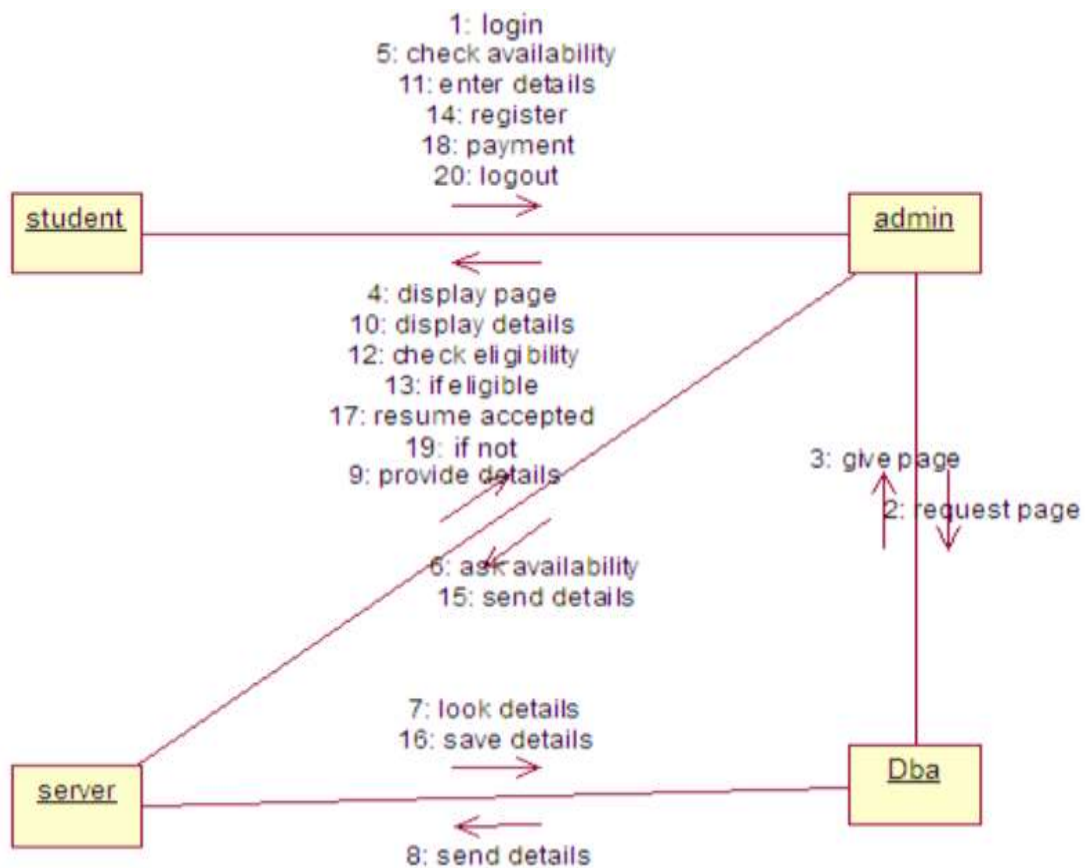


3.3 Structural Analysis

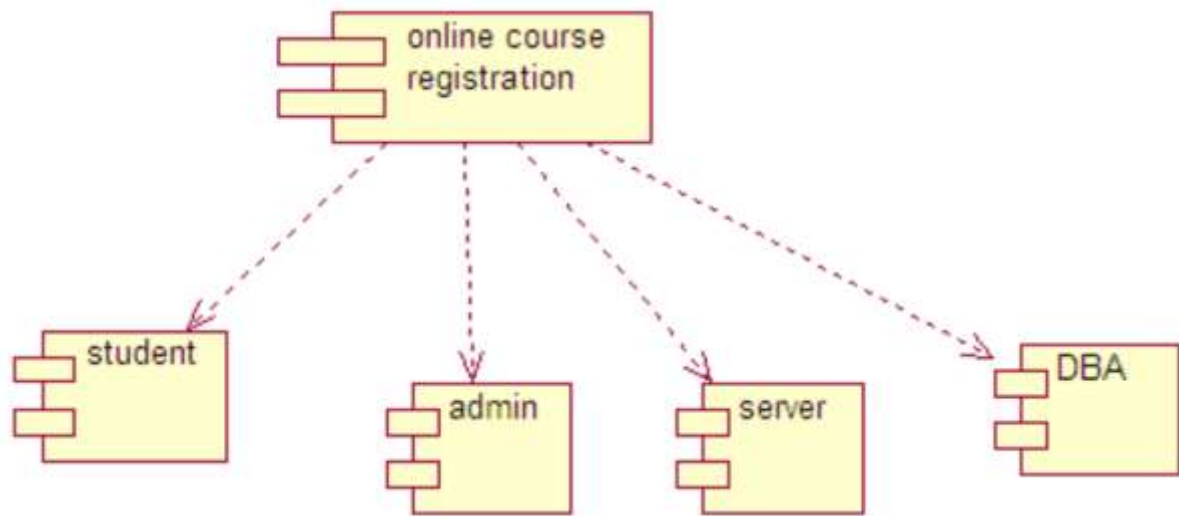
Deployment diagram:



Collaboration diagram:



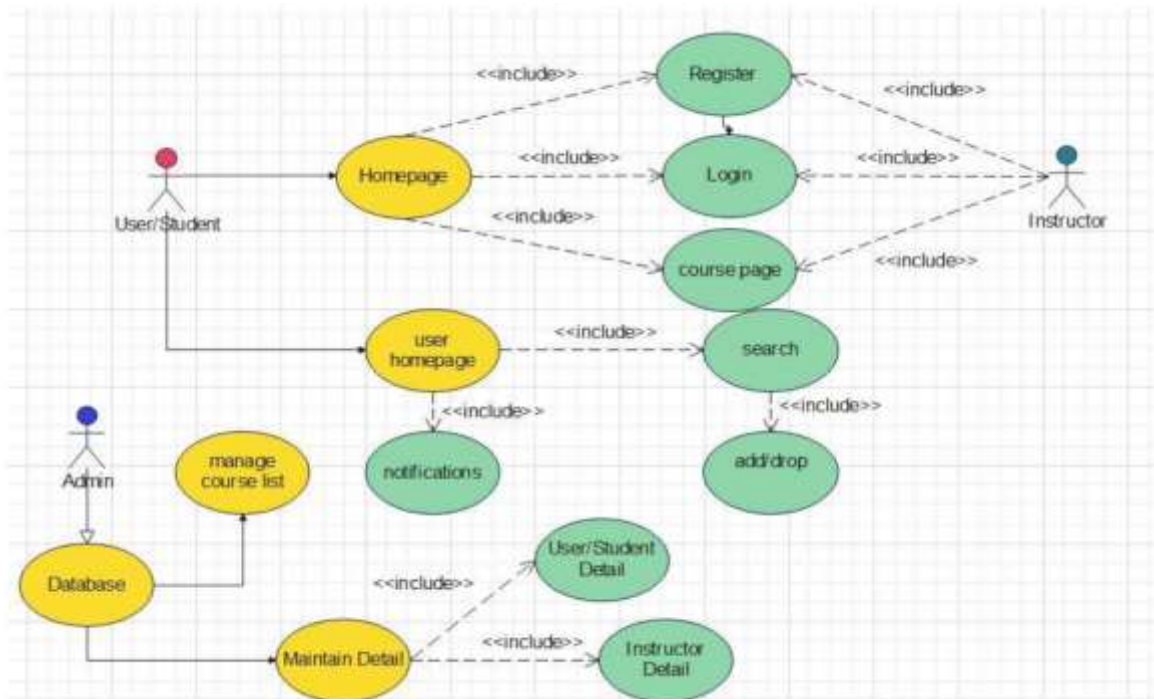
Component diagram:



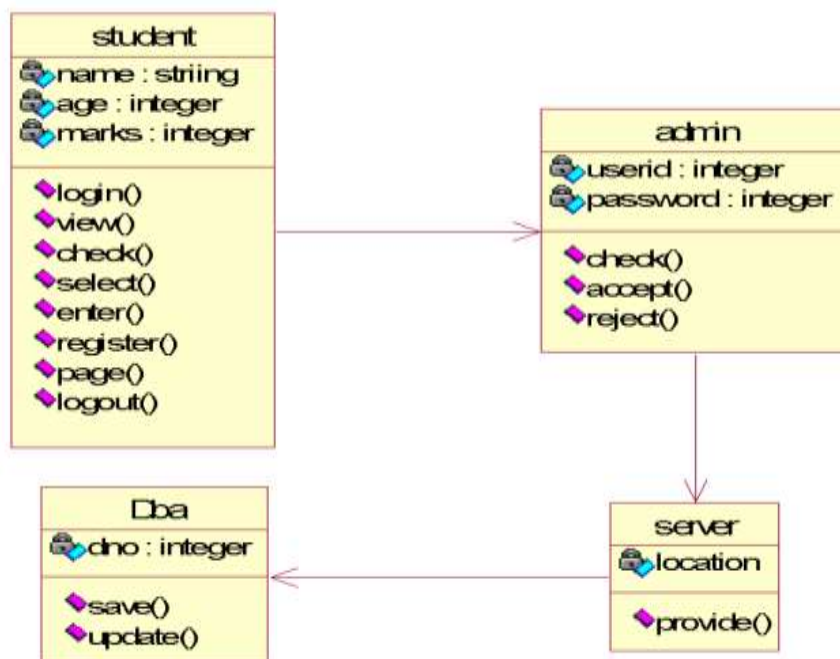
Chapter 4 Designs

4.1 UML Designs

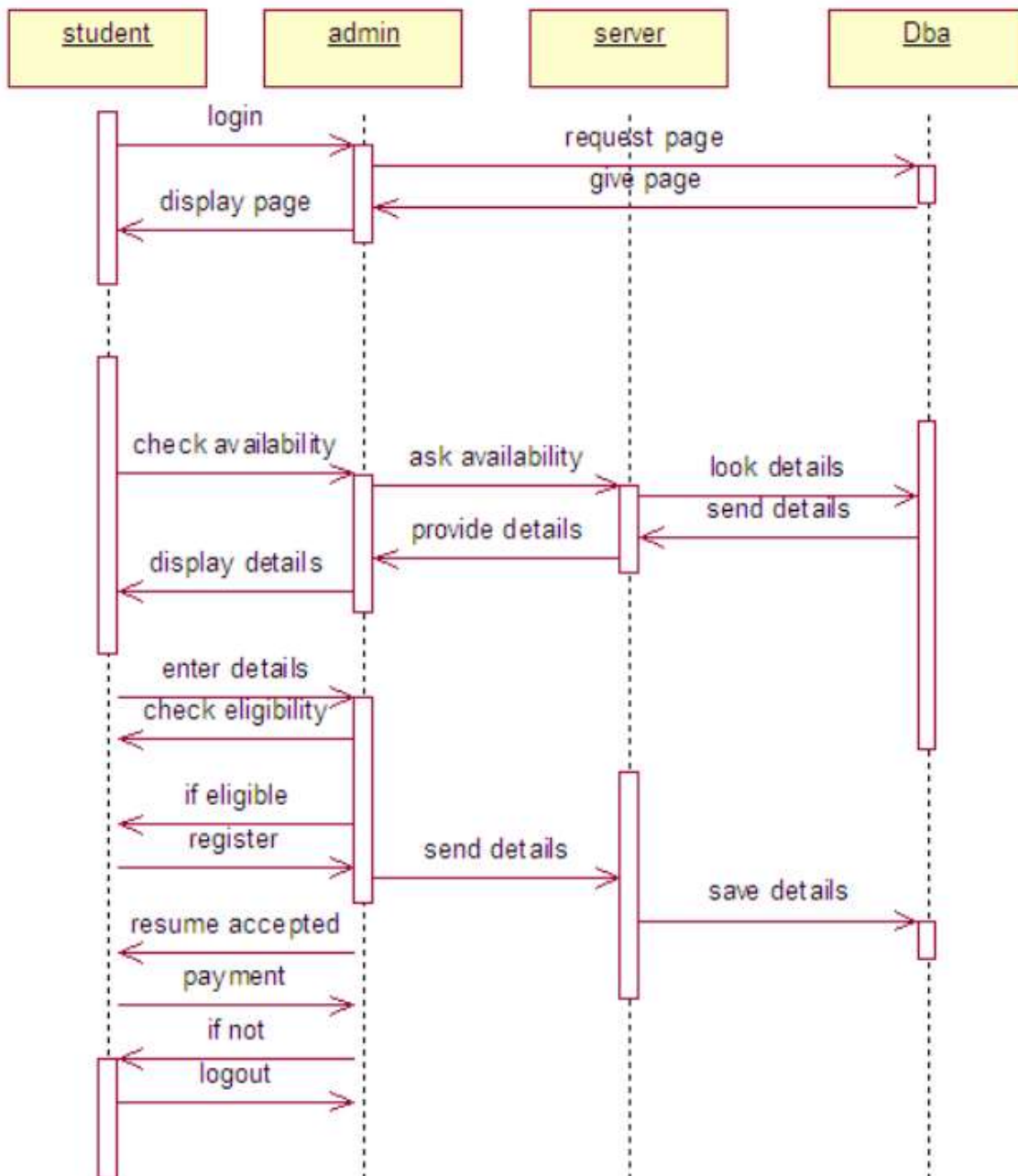
Use case diagram:



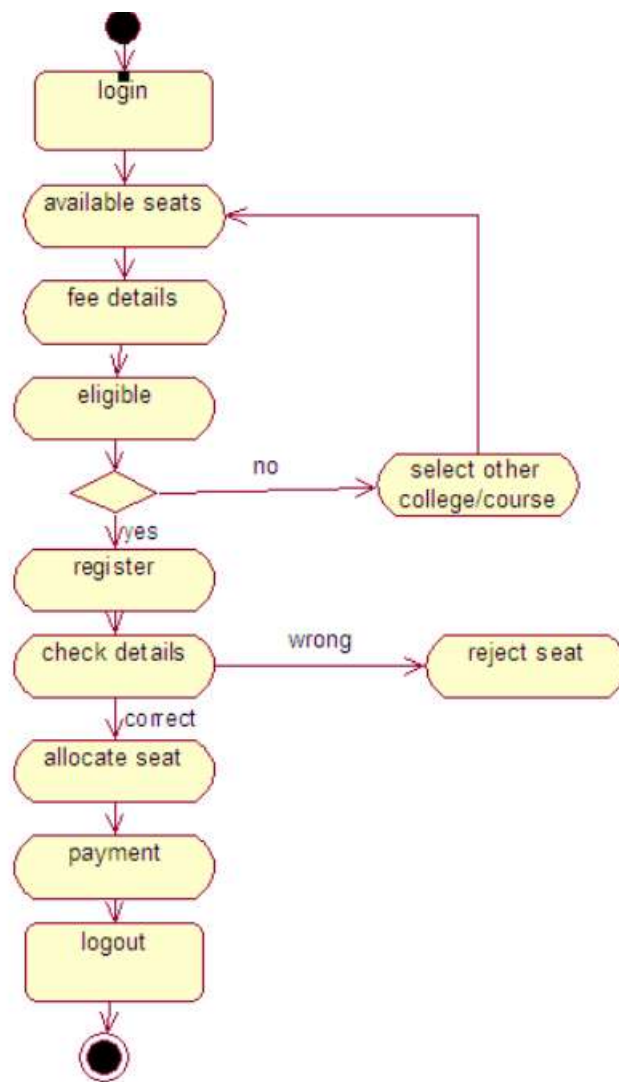
Class diagram:



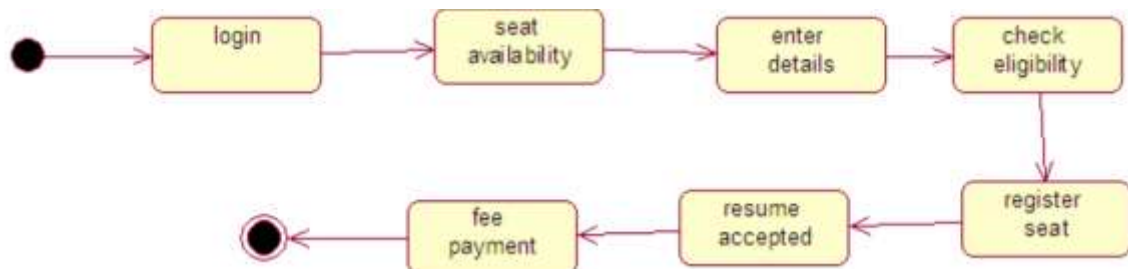
Sequence diagram



Activity diagram:



State diagram:



4.2 Algorithms and Flow Charts

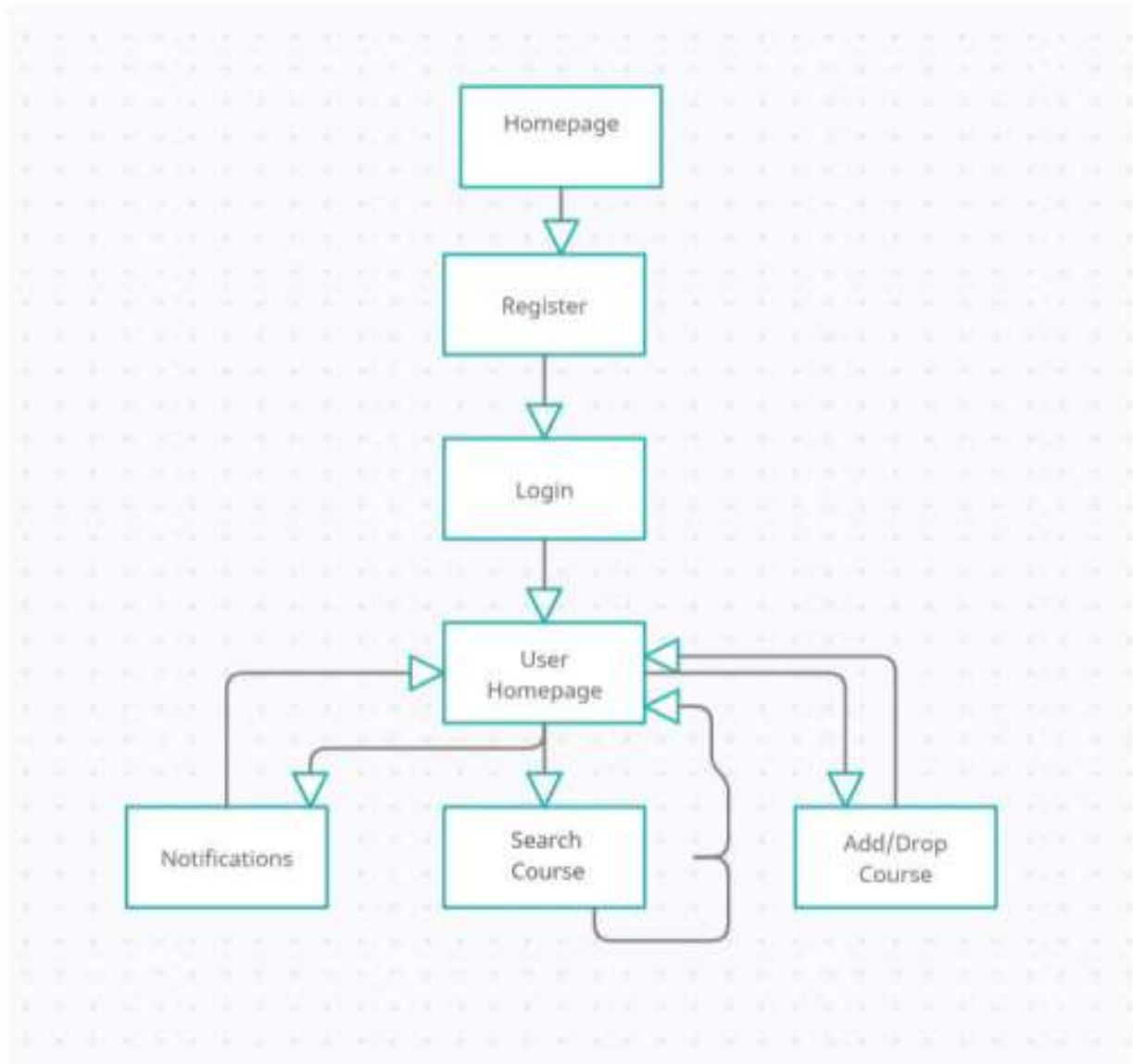


Fig 2: flow diagram

St-1: Click “Register” Button: Account Registration

6. The system shall allow non-registered user to create a secure account.
7. The system shall require the following information: First name, Last name, E-mail address.
8. The system shall ask the user for student number as a username and password.

9. The system shall confirm the username and password are acceptable
10. The system shall store the information in the database

St-2: Click “Login” Button: Account Login

4. The system shall allow a registered user to log-in to their account.
5. The system shall require a username and password from the user.
6. The system will verify the username and password, and the user will be considered “logged-in”.

St-3: Click “Registration” Button: Course Registration

8. The system shall allow users to register for courses offered.
9. The system shall require user’s personal formation such as: Full name, Identity number, date of birth, Address, etc.
10. The system shall require the user to specify the level of study and allow the user to select courses to register for.
11. The system shall confirm all valid courses offered for the level of study selected by user.
12. The system shall also require the user to provide credit information.
13. The system shall allow user to save the already entered details for later continuation.
14. After the user filled the entire form, the system shall save the data in the database.

St-4: Click “Cancel Registration” Button: Cancel

3. The system shall allow the user to cancel the entire registration.
4. Alternately the user may wish to cancel while filling the form.

St-5: Click “Replace Course” Button: Replace

4. The system shall allow user to replace one with another.
5. This can happen while or after registration.
6. The system shall be given option to progress back to invalid courses entered.

St-6: Click “View available Courses” Button: View Courses

3. The system shall allow user to view available courses before registration.
4. The system shall view available space left for each course.

St-7: Click “submit registration “Button: submit

3. After filling in the registration form, the system shall allow user to submit the registration.
4. After submitting a copy of the proof of registration will appear on the screen.

St-8: Click “view registration status” Button: view status

3. System shall allow user to click and view their registration status. This includes the courses registered and proof of registration.
4. System shall allow user to print the proof of registration

Chapter 5 Development

5.1 Tools Description and Development approach used

Hardware Interfaces

This system will not be engaging with the hardware in terms of exchanging information or functionalities, it will run on the hardware. The system will be dependent on the capabilities of the hardware and will not alter any functionalities of the hardware

Minimum Hardware Requirements:

The system will run on different hardware gadgets. Below are the minimum hardware requirements for the smooth running of the system:

- 1GB RAM PC
- 1.8Hz processor
- 14" color monitor
- 120GB HDD CPU
- Proper running internet

Software Interfaces

The system will require a lot of software connections so it has to be compatible to external software to be able to connect. The main system will be composed of the web portal, the operating system used and the database all communicating together and exchanging information.

Below are the different software components that the system will need to be fully operational under their respective section headings:

Databases

The following databases can be used for the system

- MongoDB
- MySQL

Operating systems

The system will be compatible to the following operating systems:

- Windows7
- Linux
- Mac OS

Tools

The following tools will be used for the system's development:

- Eclipse
- Notepad++
- HTML 5
- Node.js
- JavaScript
- CSS
- React

Communications Interfaces

The main communication link that the system will be using is the internet. Below is the different information transferring methods and their respective protocols:

Information transmission means:

- Web file transfer
- E-mail
- Social Networks information transfer

Protocols

The protocols that are to be used in this system are the secure ones as confidentiality is of vital importance in this system. Listed below are the protocols:

- HTTPs
- FTPs
- SMTPs

5.2 Pseudocodes of Important Modules.

Register

```
app.post("/Register", async (req, res) => {  
  try{  
    console.log(req.body)  
    const userd= new User({  
      fullname: req.body.fullname,  
      username: req.body.username,  
      email: req.body.email,  
      password: req.body.password,  
      gender: req.body.gender,  
      interests: req.body.interests,  
    })  
    const registered = await userd.save()  
    ();  
    if(registered===undefined)  
    {  
      const msg = {"success":false}  
      res.send(msg);  
    }  
    else  
    {  
      const msg = {"success":true}  
      res.send(msg);  
    }  
    // console.log(registered);  
  }  
})
```

Login

```
app.post("/Login", async (req, res) => {  
  
  try{  
    console.log(req.body);  
    const userd= await User.findOne({username:req.body.username});  
    console.log(`Found username ${userd}`);  
  
    if (req.body.password=== userd.password)  
    {  
      console.log("Sucessful login");  
      const username = req.body.username;  
      console.log(username);  
      // const token =jwt.sign({username},process.env.TOKEN_SECRET);  
      // res.status(200).send({ auth: true, token: token });  
      console.log(userd.password);  
      const senddata = {"id":userd._id,"success":true};  
      console.log(senddata);  
      res.send (senddata);  
    }  
    else  
    {  
      const msg = {"success":false};  
      res.send(msg);  
    }  
  }  
})
```

```

app.post("/users", async (req, res) =>
{
  // console.log(req.body);
  try{
    // console.log(req.body);
    const userd= await User.findOne({_id:req.body.id});
    // console.log(`Found username ${userd}`);
    const username = userd.username;
    const course = userd.courses;
    const interests = userd.interests;
    const senddata = {"username":username,"courses":course,"interests":interests};
    // console.log(senddata);
    res.send (senddata);
  }
  catch(e){
    console.log('error');
    res.send('Error');
  }
})

```

Nav

```

1  import React,{useState} from 'react';
2  import './Navbar.css';
3  function Navbartext() {
4
5      const [usersearch,setusersearch] = useState ();
6      const handleinput = (e) =>
7      {
8          const value = e.target.value;
9          setusersearch(value)
10     }
11     const loghandlere =(e)=>
12     {
13         const dend = {"course":usersearch}
14         console.log(dend);
15         e.preventDefault();
16         fetch('http://localhost:8000/Search',{
17             mode: 'cors',
18             method : 'POST',
19             headers : {
20                 'Accept' : 'application/json',
21                 'Content-type' : 'application/json'
22             },
23             body:JSON.stringify(dend)
24         }).then((result)=>result.json()).then(data=>{bkkk(data)})
25     }
26     const bkkk = (data) =>{
27         localStorage.setItem("course",JSON.stringify(data));
28         window.location.replace("/searchresults");
29     }
30     return(
31     <div className = "navbar">
32         <form className="search-bar" onSubmit={loghandlere}>
33             <input type="text" name="usersearch" id="usersearch" defaultValue={usersearch} onChange={handleinput}

```

JS Register.js M X

Frontend > Online-Course > src > JS Register.js > ...

```
1 import React,{useState} from 'react';
2 import './App.css'
3 const stylesheet ={Title:{fontFamily:"Times"}}
4 function Register () {
5   const [array,setarray] = useState([]);
6   const [userdetails,setuserdetails] = useState ({
7     fullname:"",
8     username:"",
9     password:"",
10    email:"",
11    gender:"",
12    interests:[]
13  });
14  const handleinput = (e) =>
15  {
16    const name = e.target.name;
17    const value =e.target.value;
18    setuserdetails({...userdetails,[name]:value})
19  }
20  const handleinputs = (e) =>
21  {
22    const value = e.target.value;
23    const check = e.target.checked;
24    const temp = [...array];
25    if (check === true)
26    {temp.push(value);
27    // array.push(value);
28    }
29    else if (check == false)
30    {
31
32      temp.splice(temp.indexOf(value),1)
33    }
```


Chapter 6 Testing

6.1 Test cases for Important Modules

| Requirement | Test Case ID | Test Case | Expected Result | Actual Result | Test Outcome |
|--------------|--------------|---|---|--|--------------|
| Register | 1 | User tries to register and enter details all correctly. | Redirect to login page and save the user in database | The user details have been saved and they have been redirected | Passed |
| | 2 | User tries to register with incorrect details | Redirect to register page and make user enter details again | Error was occurred in the backend and user details was not saved | Passed |
| Login | 3 | User tries to login with right credentials | Redirect to User Home page | Backend sends okay signal and user is redirected to the required page | Passed |
| | 4 | User tries to login with invalid credentials | Redirect to Login page | Error occurs in backend and sends bad signal which causes user to stay in login page | Passed |
| UI/Home page | 5 | User opens the opening website | The GUI and stylings are working properly | All of the graphical components are working fine | Passed |

| | | | | | |
|-----------------|----|--|---|---|--------|
| | 6 | User opens the user Homepage by login | Shows course list and show message when user has not added any course | The user course list is being shown properly and messages are shown to the user accordingly | Passed |
| Add/Drop Course | 7 | User clicks the add course button | The course is added and shown on user home screen | Course is added to the user database and it shows in user homepage | Passed |
| | 8 | User clicks the drop course button | The course is dropped and is removed from user screen | Course is removed from the user database and it is not shown in user homepage | Passed |
| Search | 9 | The user searches course with name matching or having starting letters same as course in the databases | The user is then redirected to a page and shown option to add those courses he searched for | The search result is shown from the database and shown to the user and add option is also there | Passed |
| | 10 | The user doesn't search course with name matching or having starting letters same as course in the databases | In search result page, message is shown that there is no course by that name | Backend returns an empty array and it is shown to the user that no course is in the database. | Passed |

| | | | | | |
|--------------|----|---|--|---|--------|
| Notification | 11 | User logs in to the page and keeps on using the website | A popup is shown at the bottom which shows the course users might like | A random course is sent from the backend and a popup screen is shown for the user | Passed |
| | | | | | |

Chapter 7 Screenshots of Developed Product



